

## Remarks

Claim 7 has been amended to recite that the activation module instance “running in said second mode” runs as a daemon, as recited in the specification at, for example, page 10, lines 11-12. With this amendment, claim 7 and claim 8 dependent thereon are believed to avoid any ground for rejection under 35 U.S.C. § 112, second paragraph.

The Examiner’s indication of allowability of claims 5 and 7-10 if rewritten in independent form is noted with appreciation. Except for claim 7, however, applicant has left these claims in their original form out of a belief that the claims upon which they depend are similarly allowable.

Claims 1-2, 6, 12 and 14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Deianov et al. 6,529,985 (“Deianov”) in view of Hammond 6,463,583, alone or in conjunction with other art (paper no. 3, ¶¶ 8 and 16). These rejections are respectfully traversed.

Claim 1, upon which the remaining claims depend, is directed to a system in which, among other features, an activation module is adapted to load an interception module to occupy a location in a shared region of virtual memory as long as interception of API calls is required.

Deianov discloses a system for selective interception of API calls. As shown in Fig. 1 of the patent, one embodiment of the system includes an interception module 111 and an interrupt vector table 113 in operating system (OS) address space 105, together with a modified loader program 121 and a system call wrapper 125 in user address space 103. (In other embodiments, the system call wrapper 125 may be in OS address space.) In the interrupt vector table 113, selected pointers 114 to system calls 115 are replaced with pointers 118 to the interception module 111.

As the Examiner concedes, Deianov does not teach loading the interception module to occupy a location in a shared region of virtual memory as long as interception of the API calls is required, as claimed by applicant (paper no. 3, ¶ 10). The Examiner asserts, however, that Hammond teaches this feature and that it would have been obvious to combine the two references because

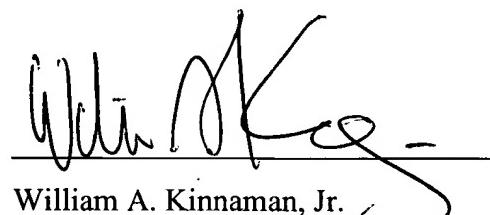
Hammond's teaching of loading the interception module "would improve the flexibility of Deianov's system by dynamically injecting the execution logic into a shared memory space of a window[ed] operating system" (paper no. 3, ¶ 11). Applicant respectfully disagrees.

Contrary to the Examiner's assertion, Hammond does not teach loading an interception module to occupy a location in a shared region of virtual memory as long as interception of the API calls is required, as claimed by applicant. Rather, Hammond discloses a system for dynamically injecting execution logic in the form of a dynamic link library (DLL) into a shared memory space of a windowed operating system (col. 2, lines 56-58; col. 8, line 36 to col. 9, line 3). Other than the fact that it loads an executable module in a shared user address space (which is certainly not new), Hammond has nothing to do with either Deianov or applicant's claimed invention. Certainly it would not suggest loading Deianov's interception module 111 into such shared user space when that reference so repeatedly teaches that it be loaded into the OS address space 105 (e.g., col. 5, line 64 to col. 6, line 2; col. 10, lines 26-30; col. 11, lines 54-58).

For the foregoing reasons, claim 1 and the claims dependent thereon are believed to distinguish patentably over the art cited by the Examiner. Reconsideration of the application as amended is respectfully requested. It is hoped that upon such consideration, the Examiner will hold all claims allowable and pass the case to issue at an early date. Such action is earnestly solicited.

Respectfully submitted,  
RICHARD JOHN MOORE

By



William A. Kinnaman, Jr.  
Registration No. 27,650  
Phone: (845) 433-1175  
Fax: (845) 432-9601

WAK/wak